THE CRAIG WEISS
GROUP

BASIC TO EXPERT • IN PLAIN ENGLISH

The AI Terminology Guide

Every AI term worth knowing, explained the way you'd explain it to a friend at lunch — not the way a textbook would. Starts easy. Gets sharper. No filler.

Why Bother Learning the Lingo

Because the jargon is doing a lot of gatekeeping it doesn't need to do

Every industry does this. Somebody invents a genuinely useful thing, and within about a year, a wall of jargon springs up around it — half of it necessary, half of it just people showing off in meetings.

AI's no different. "Tokens." "Embeddings." "Inference." None of it is actually complicated once somebody just tells you straight. So that's what this is. Every term worth knowing, in order — starting basic, ending sharp — with a real example under each one so it actually sticks.

Skip around if you want. Read it front to back if you want. Either way, by the end you'll know more than most people in the room. That's not a high bar. Doesn't matter. Still worth clearing.

Level 1: The Basics

Start here if any of this is new to you

Artificial Intelligence (AI)

The umbrella term. Software that does things that used to require an actual human brain — reading, writing, recognizing a face, holding a conversation. Not magic. Very good pattern-matching, dressed up nice.

Example: Spam filters, Netflix recommendations, and the thing writing this sentence right now. All AI. Wildly different jobs.

Machine Learning (ML)

How AI actually learns. Instead of a person hand-coding every rule, you feed the system a mountain of examples and it works out the pattern on its own. Nobody wrote 10,000 rules for "what a cat looks like." The machine figured it out.

Example: Show a system a million photos labeled cat or not-cat, and eventually it gets scary good at telling the difference.

Generative AI

The kind of AI that creates something new — text, images, code, video — instead of just sorting or predicting. This is the stuff that took over the last few years. Ask it for a poem about your dog, it writes one. Nobody wrote that poem in advance.

Example: ChatGPT drafting an email, Midjourney painting a picture, Claude writing this guide. All generative.

Large Language Model (LLM)

The engine under the hood of most text-based AI you use. Trained on an enormous pile of text, so it's gotten very good at predicting the next word, and the next, and the next — which, done well enough, starts looking a lot like understanding.

Example: Claude, ChatGPT, Gemini — all LLMs, each one wearing a different friendly interface.

Small Language Model (SLM)

The other side of the LLM coin. Same basic idea — trained on text, predicting language — just built small on purpose. Fewer parameters, less horsepower needed, cheaper and faster to run. Not weak by accident. Traded some general, know-everything ability for speed, cost, and the option to run on smaller hardware instead of a data center.

Example: A phone that can summarize your texts without sending anything out over the internet is likely running a small language model right there on the device, not calling out to a giant one somewhere else.

Chatbot

The oldest, least fancy word on this whole list. Anything you type messages to and it types back. Doesn't mean it's smart. That 2015 customer service bot that only understood "YES" or "NO" was technically a chatbot too.

Example: A modern LLM assistant is a chatbot. So was that clunky airline bot from a decade ago. Same category, very different quality.

Prompt

What you type. Your instructions, your question, your request — however you phrase it. Good prompt, good answer. Vague prompt, vague answer. We've said this before. It's still true.

Example: "Write me a birthday email" is a prompt. So is "Write a birthday email for my aunt Carol, warm, under 100 words." Guess which one works better.

Token

The chunk of text AI reads and writes in — roughly three-quarters of a word. Vendors bill you on these. If the bill's the part that worries you, we wrote a whole guide on that.

Example: "Hello" is one token. "Unbelievable" might get split into two or three.

Training Data

Everything the model read before you ever typed a single word to it. Books, articles, code, conversations — whatever the company fed it to learn from. The model doesn't know anything past that cutoff, which is why it can be behind on current events.

Example: If training stopped in early 2025, don't ask it about something that happened in late 2025. It won't know. That's not a bug.

GOOD TO KNOW

None of these four terms are optional if you want to sound like you know what you're talking about in a meeting: AI, LLM, prompt, token. Get those down cold before moving on.

Level 2: Getting Sharper

For people who've been using this stuff a while

Natural Language Processing (NLP)

The whole field of teaching computers to deal with human language — reading it, understanding it, generating it. LLMs are the current superstar of NLP, but NLP existed in much dumber forms long before any of this.

Example: Spell-check is NLP. So is a modern AI writing your quarterly report. Same family, very different sophistication.

Context Window

How much the AI can "remember" at once, in a single conversation. Go past it, and older stuff starts falling out the back, like a bucket that's already full.

Example: Paste in a 300-page document and ask about chapter 1 after 250 pages of back-and-forth. If you're past the window, it may have already forgotten chapter 1.

Fine-Tuning

Taking an already-trained model and giving it extra, focused training on a narrower set of data so it gets sharper at one specific job. Hiring a smart generalist, then sending them to a two-week bootcamp in your industry.

Example: A general model fine-tuned on thousands of legal contracts gets noticeably better at contract review than the stock version.

Hallucination

When the AI confidently makes something up. Not lying, exactly — it doesn't know it's wrong. It's predicting what sounds right, and sometimes what sounds right isn't true. This is why you check facts, dates, and sources. Every time. No exceptions.

Example: Ask for a citation and get a real-sounding book title and author that simply doesn't exist. It happens. Check your sources.

Model

The actual trained system doing the work. When people ask "which model are you using," they mean which specific AI brain — not which app, not which company.

Example: Claude Opus and Claude Sonnet are two different models from the same company — one bigger and slower, one leaner and faster.

API

The plumbing. A way for one piece of software to talk to another without a person clicking buttons in between. Companies build products on top of AI through an API, instead of you typing into a chat box.

Example: A scheduling app that quietly drafts your meeting summaries is probably talking to an AI model through an API — no chat window involved.

Multimodal

Handles more than just text. Images, audio, video — a multimodal model can take in or produce more than one kind of content.

Example: Uploading a photo and asking "what's wrong with this spreadsheet screenshot" — that's multimodal. A text-only model can't touch that.

Temperature

A setting controlling how safe or wild the AI's answers get. Low temperature, predictable and consistent. High temperature, more creative — and more likely to go off the rails. Most people never touch this. Now you know it's there.

Example: Low temperature for a legal summary. High temperature if you want ten weird, creative taglines to pick from.

Tokenmaxxing

Not an official term. Built the same lazy way "looksmaxxing" got built — tack "maxxing" onto whatever people are chasing to the extreme. Tokenmaxxing means designing a prompt, a workflow, or an entire product around burning through as many tokens as possible, either because someone assumes more tokens equals a better answer, or because the billing structure quietly rewards it. Companies need to break this habit, and fast: past a certain point, more tokens doesn't mean a better answer. It just means a longer one — padded, repetitive, saying the same thing three different ways. Products built around tokenmaxxing (chatty system prompts, dumping whole databases into every request instead of retrieving what's actually needed) aren't buying quality. They're buying a bigger bill. If your AI costs are climbing faster than your quality, this is usually why.

Example: A support bot that pastes your entire 40-page policy manual into every single question, instead of pulling the one relevant paragraph. Same answer. Ten times the cost.

AI Slop

The stuff that gets made because generating it costs almost nothing, not because anyone needed it. Low-effort text, images, or video, cranked out in bulk and dumped online with little to no human judgment applied on the way out. The name says it plainly: nobody's proud of it, it's just filling space. Why it's detrimental: it drowns out the good stuff. Search results, social feeds, even inboxes get buried under content nobody checked, nobody edited, and often nobody even read before hitting publish — which makes it harder for anything genuinely useful to get found, and slowly trains people to trust everything a little less, including the AI-assisted work that's actually good.

Example: A "10 Best Vacation Spots" article, clearly never edited by a human, listing a city that doesn't exist and repeating the same sentence structure eleven times. Technically words. Not actually useful.

Level 3: Now You're Dangerous

The terms that separate a casual user from someone who actually gets it

Retrieval-Augmented Generation (RAG)

A method where the AI looks something up in an outside source — a document, a database — before answering, instead of relying purely on what it memorized during training. Keeps answers current and grounded in your actual material, not just its general knowledge.

Example: A company chatbot that searches your internal policy documents before answering an HR question, instead of guessing from general knowledge.

Embeddings

A way of turning words, sentences, or documents into numbers that capture meaning, so a computer can measure how similar two pieces of text actually are. This is the math underneath search, recommendations, and RAG.

Example: "Dog" and "puppy" end up mathematically close together. "Dog" and "stapler" do not. That closeness is the embedding doing its job.

Reasoning / Chain-of-Thought

When a model works through a problem step by step instead of jumping straight to an answer — the AI version of showing your work in math class. Slower, often noticeably more accurate on genuinely hard problems.

Example: A tricky logic puzzle solved by walking through each clue in order, instead of blurting out a guess.

Agentic AI / AI Agents

AI that doesn't just answer — it acts. Takes steps, uses tools, completes multi-part tasks with less hand-holding. Different animal entirely from a chatbot that only replies when spoken to.

Example: An agent that researches five vendors, compares pricing, and drafts a comparison document — without you babysitting every step.

MCP (Model Context Protocol)

A standard way for an AI model to connect to outside tools and data — your calendar, a database, a company's internal software — without a developer hand-building a custom connection for every single pairing. Before something like this existed, hooking an AI up to twenty different tools meant building twenty different one-off bridges. MCP is an attempt at one bridge design everybody can reuse, instead of reinventing it each time.

Example: An AI assistant that checks your calendar, pulls a file from your drive, and creates a task in a project tracker, all in one conversation, is very likely using something like MCP to talk to each of those apps behind the scenes.

Guardrails / Alignment

Guardrails are the rules baked into a system to keep it behaving — staying safe, staying honest, staying inside the lines a company set. Alignment is the bigger research question behind it: making sure AI actually does what humans want, not just what it was literally told.

Example: A model refusing to help with something dangerous, even when asked cleverly. That refusal is a guardrail doing its job.

Inference

The actual moment of the AI generating a response to you — as opposed to "training," which already happened, long before you typed anything. When people talk about the day-to-day cost of running AI, they usually mean inference costs.

Example: Training a model might take months and a small country's worth of electricity. Inference — answering your one question — takes seconds.

Prompt Injection

The one term on this list worth slowing down for

Prompt injection is when someone sneaks hidden instructions into content an AI is about to read, hoping the AI follows the hidden instructions instead of doing the job it was actually given.

It works because most AI systems don't have a built-in way to tell the difference between "the actual content I'm supposed to review" and "a command I'm supposed to obey." Text is text, as far as the model's concerned — unless the system around it was built carefully to keep those two things separate. A lot of them weren't. Not yet, anyway.

Why This Should Worry Anyone Using AI to Screen Job Applications

Here's where it gets uncomfortable. Say a company's AI recruiting tool scans resumes and rates candidates automatically. Somebody who knows the resume is going through an AI, not a human eyeball, can hide extra instructions inside the document — text small enough or colored close enough to the background that a person skimming it never notices. An instruction telling the AI reading it to treat this candidate as a strong match, overlook any gaps, rank them near the top.

A human reviewer never sees it. But if the screening tool wasn't built to tell the difference between "content to evaluate" and "instructions to follow," it might just read that hidden text and comply. No hacking involved. No special skill beyond knowing how to change a font color.

Why It Actually Bypasses the Screener

Because most of these tools were built to save time, not to withstand someone actively trying to game them. The screener's whole job is to read a document and produce a score. If it can't separate the resume's actual content from an instruction buried inside that content, a single well-placed line can quietly hijack the result — completely invisibly, to everyone involved except the person who planted it.

GOOD TO KNOW

This cuts both ways. For a candidate, it's tempting and it's cheating — and it can absolutely backfire if anyone checks. For a company, it's a real argument against trusting an automated resume screen with zero human review. If you're the one buying AI recruiting software, ask the vendor directly: how do you separate instructions from content submitted by an applicant? If they don't have a clean answer, that's your answer.

This isn't just an HR problem, by the way. Same trick, same logic, works anywhere an AI reads content it didn't write and is expected to act on it responsibly — emails, documents, web pages, customer messages. Prompt injection is the reason "just point AI at anything and trust it completely" is still bad advice, no matter how good the model gets.

Data Centers

The building nobody thinks about until they have to

What It Actually Is

A data center is a building full of computers doing the actual work. Every time you type something into any AI, that request isn't answered by a chip inside your phone or laptop. It goes out over the internet to a warehouse-sized building somewhere, packed floor to ceiling with servers, and one of those servers processes your request and sends the answer back. Feels instant. Isn't happening anywhere near you.

What It Does

Stores data, runs the software, and — for AI specifically — handles both training (building the model in the first place) and inference (answering your actual questions), on hardware far more powerful than anything sitting on your desk. Specialized chips, mostly, built for exactly this kind of math, running around the clock, every day of the year.

The Environmental Impact — the Part Getting Real Attention Lately

And for good reason. These buildings use enormous amounts of electricity. Training a large model can burn through as much power as a small town uses in a year, and running the finished model for millions of people afterward keeps that number climbing, not falling. They also need serious amounts of water, used to keep all that hardware from overheating. Depending on where a data center sits and what's powering the local grid, that electricity comes from clean sources, dirty ones, or some mix of both — which is exactly why companies are fighting over where to build these things, and increasingly investing directly in nuclear or renewable power just to keep up with demand.

PROS VS. CONS, PLAINLY

Pro: this is what makes AI possible at any real scale. Try running a large model on your laptop — it's not happening, or it's happening too slowly to be useful. Centralizing the horsepower is what lets millions of people use the same system at once.

Con: the environmental cost is real, it's growing, and it's not evenly distributed. Some communities end up with strained power grids and water supplies because a data center moved in next door — and see approximately none of the benefit.

Bottom line: data centers aren't going anywhere as long as AI keeps growing. The honest question isn't whether they have an environmental impact — they clearly do. It's whether a given company is being straight about the size of that impact and actually investing in cleaner power to offset it. Some are. Some are mostly doing it for the press release. Worth knowing the difference before taking a vendor's sustainability claims at face value.

Vendor Claims Worth a Second Look

Fact-checking the pitch, one claim at a time

Vendors love a confident claim. "Our RAG reduces hallucinations." "Our guardrails keep the model accurate." "Our new model makes fewer mistakes." All three get repeated constantly, in every sales deck. None of them are entirely false. None of them are the whole story either. Let's sort out what's actually true.

The Claim: “More RAG Means Fewer Mistakes”

Is it true? Mostly, yes — with a real asterisk attached. RAG grounds the model's answer in an actual document instead of whatever it half-remembers from training, and that genuinely does cut down on one specific kind of mistake: making up facts that sound plausible but aren't real. That part's legitimate.

The asterisk: RAG is only as good as what it retrieves. Feed it outdated, wrong, or poorly organized source material, and it will confidently repeat that bad information right back to you — now with a source attached, which paradoxically makes it look more trustworthy, not less. More RAG doesn't fix bad data. It just launders it.

The Claim: “More Guardrails Mean Fewer Mistakes”

Is it true? Not really, and this one gets conflated constantly. Guardrails are mostly about safety and behavior — keeping a model from saying something harmful, off-brand, or outside its lane. That's a real, valuable job. But guardrails don't make a model smarter or more factually accurate. A heavily guardrailed model can still hallucinate a statistic with total confidence. Different problem. Different fix. Don't buy "guardrails" as an answer to "is it accurate."

The Claim: “A Bigger, Newer LLM Means Fewer Mistakes”

Is it true? Generally, yes, on average, across benchmarks — newer models do tend to make fewer factual errors than their predecessors. That part checks out. Where it gets oversold is the word “fewer.” Fewer isn't “none.” Every model, including the newest, most expensive one on the market, will still confidently say something wrong sometimes. “Better” is real. “Solved” is marketing.

GOOD TO KNOW

None of these three are useless. RAG, guardrails, and bigger models all genuinely help, each with a different kind of problem. The mistake is treating any one of them as the whole answer to “make sure the AI doesn't mess up.” It takes the combination — plus a human somewhere in the loop actually checking the output, especially anywhere a mistake would cost you something real.